CHAPTER 51

Configuring SNMP

This chapter describes how to configure the Simple Network Management Protocol (SNMP) on the Catalyst 4500 series switch.

Note For complete syntax and usage information for the commands used in this chapter, see the *Cisco IOS Configuration Fundamentals Command Reference*

http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html

and related publications at this location:

http://www.cisco.com/en/US/products/ps6350/index.html

This chapter consists of these sections:

- Understanding SNMP, page 51-1
- Configuring SNMP, page 51-5
- Displaying SNMP Status, page 51-17

Understanding SNMP

SNMP is an application-layer protocol that provides a message format for communication between managers and agents. The SNMP system consists of an SNMP manager, an SNMP agent, and a MIB. The SNMP manager can be part of a network management system (NMS) such as CiscoWorks. The agent and MIB reside on the switch. To configure SNMP on the switch, you define the relationship between the manager and the agent.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value into the agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager's requests to get or set data.

An agent can send unsolicited traps to the manager. Traps are messages alerting the SNMP manager to a condition on the network. Traps can mean improper user authentication, restarts, link status (up or down), MAC address tracking, closing of a Transmission Control Protocol (TCP) connection, loss of connection to a neighbor, or other significant events.

This section includes information about these topics:

- SNMP Versions, page 51-2

- SNMP Manager Functions, page 51-3
- SNMP Agent Functions, page 51-4
- SNMP Community Strings, page 51-4
- Using SNMP to Access MIB Variables, page 51-4
- SNMP Notifications, page 51-5

SNMP Versions

The Catalyst 4500 series switch supports these SNMP versions:

- SNMPv1—The Simple Network Management Protocol, a Full Internet Standard, defined in RFC 1157.
- SNMPv2C replaces the Party-based Administrative and Security Framework of SNMPv2Classic with the community-string-based Administrative Framework of SNMPv2C while retaining the bulk retrieval and improved error handling of SNMPv2Classic. It has these features:
  - SNMPv2—Version 2 of the Simple Network Management Protocol, a Draft Internet Standard, defined in RFCs 1902 through 1907.
  - SNMPv2C—The community-string-based Administrative Framework for SNMPv2, an Experimental Internet Protocol defined in RFC 1901.
- SNMPv3—Version 3 of the SNMP is an interoperable standards-based protocol defined in RFCs 2273 to 2275. SNMPv3 provides secure access to devices by authenticating and encrypting packets over the network and includes these security features:
  - Message integrity—ensuring that a packet was not tampered with in transit
  - Authentication—determining that the message is from a valid source
  - Encryption—mixing the contents of a package to prevent it from being read by an unauthorized source.

Note To select encryption, enter the **priv** keyword. This keyword is available only when the crypto (encrypted) software image is installed.

Both SNMPv1 and SNMPv2C use a community-based form of security. The community of managers able to access the agent's MIB is defined by an IP address access control list and password.

SNMPv2C includes a bulk retrieval mechanism and more detailed error message reporting to management stations. The bulk retrieval mechanism retrieves tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2C improved error-handling includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes in SNMPv2C report the error type.

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy set up for a user and the group within which the user resides. A security level is the permitted level of security within a security model. A combination of the security level and the security model determine which security mechanism is used when handling an SNMP packet. Available security models are SNMPv1, SNMPv2C, and SNMPv3.

The following table identifies the characteristics of the different combinations of security models and levels.

Model	Level	Authentication	Encryption	Result
SNMPv1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv2C	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv3	noAuthNoPriv	Username	No	Uses a username match for authentication.
SNMPv3	authNoPriv	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
SNMPv3	authPriv (requires the cryptographic software image)	MD5 or SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard.

You must configure the SNMP agent to use the SNMP version supported by the management station. Because an agent can communicate with multiple managers, you can configure the software to support communications using SNMPv1, and SNMPv2C, and SNMPv3 protocols.

SNMP Manager Functions

The SNMP manager uses information in the MIB to perform the operations described in Table 51-1.

Table 51-1 SNMP Operations

Operation	Description
get-request	Retrieves a value from a specific variable.
get-next-request	Retrieves a value from a variable within a table.[1]
get-bulk-request[2]	Retrieves large blocks of data, such as multiple rows in a table, that would otherwise require the transmission of many small blocks of data.
get-response	Replies to a get-request, get-next-request, and set-request sent by an NMS.
set-request	Stores a value in a specific variable.
trap	An unsolicited message sent by an SNMP agent to an SNMP manager when some event has occurred.

1. With this operation, an SNMP manager does not need to know the exact variable name. A sequential search is performed to find the needed variable from within a table.
2. The **get-bulk** command only works with SNMPv2 or later.

SNMP Agent Functions

The SNMP agent responds to SNMP manager requests as follows:

- Get a MIB variable—The SNMP agent begins this function in response to a request from the NMS. The agent retrieves the value of the requested MIB variable and responds to the NMS with that value.
- Set a MIB variable—The SNMP agent begins this function in response to a message from the NMS. The SNMP agent changes the value of the MIB variable to the value requested by the NMS.

The SNMP agent also sends unsolicited trap messages to notify an NMS that a significant event has occurred on the agent. Examples of trap conditions include, but are not limited to, when a port or module goes up or down, when spanning-tree topology changes occur, and when authentication failures occur.

SNMP Community Strings

SNMP community strings authenticate access to MIB objects and function as embedded passwords. In order for the NMS to access the switch, the community string definitions on the NMS must match at least one of the three community string definitions on the switch.

A community string can have one of these attributes:

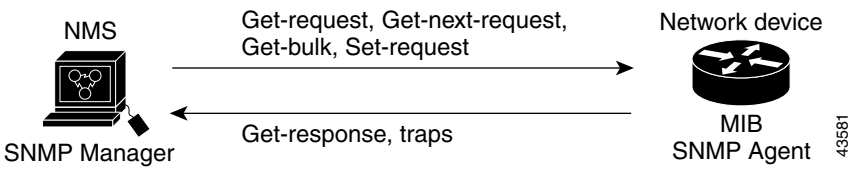
- Read-only (RO)—Gives read access to authorized management stations to all objects in the MIB except the community strings, but does not allow write access
- Read-write (RW)—Gives read and write access to authorized management stations to all objects in the MIB, but does not allow access to the community strings
- Read-write-all—Gives read and write access to authorized management stations to all objects in the MIB, including the community strings

Using SNMP to Access MIB Variables

An example of an NMS is the CiscoWorks network management software. CiscoWorks 2000 software uses the switch MIB variables to set device variables and to poll devices on the network for specific information. The results of a poll can be displayed as a graph and analyzed to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, monitor traffic loads, and more.

As shown in Figure 51-1, the SNMP agent gathers data from the MIB. The agent can send traps, or notification of certain events, to the SNMP manager, which receives and processes the traps. Traps alert the SNMP manager to a condition on the network such as improper user authentication, restarts, link status (up or down), MAC address tracking, and so forth. The SNMP agent also responds to MIB-related queries sent by the SNMP manager in *get-request*, *get-next-request*, and *set-request* format.

Figure 51-1 SNMP Network



SNMP Notifications

SNMP allows the switch to send notifications to SNMP managers when particular events occur. SNMP notifications can be sent as traps or inform requests. In command syntax, unless there is an option in the command to select either traps or informs, the keyword *traps* refers to either traps or informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

Note SNMPv1 does not support informs.

Traps are unreliable because the receiver does not send an acknowledgment when it receives a trap, and the sender cannot determine if the trap was received. When an SNMP manager receives an inform request, it acknowledges the message with an SNMP response protocol data unit (PDU). If the sender does not receive a response, the inform request can be sent again. Because they can be re-sent, informs are more likely than traps to reach their intended destination.

The characteristics that make informs more reliable than traps also consume more resources in the switch and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request is held in memory until a response is received or the request times out. Traps are sent only once, but an inform might be re-sent or retried several times. The retries increase traffic and contribute to a higher overhead on the network. Therefore, traps and informs require a trade-off between reliability and resources. If it is important that the SNMP manager receive every notification, use inform requests. If traffic on the network or memory in the switch is a concern and notification is not required, use traps.

Configuring SNMP

This section describes how to configure SNMP on your switch. It contains this configuration information:

- Default SNMP Configuration, page 51-5
- SNMP Configuration Guidelines, page 51-6
- Disabling the SNMP Agent, page 51-7
- Configuring Community Strings, page 51-7
- Configuring SNMP Groups and Users, page 51-9
- Configuring SNMP Notifications, page 51-11
- Setting the Agent Contact and Location Information, page 51-15
- Limiting TFTP Servers Used Through SNMP, page 51-15
- SNMP Examples, page 51-16

Default SNMP Configuration

Table 51-2 shows the default SNMP configuration.

Table 51-2 Default SNMP Configuration

Feature	Default Setting
SNMP agent	Enabled
SNMP trap receiver	None configured
SNMP traps	None enabled except the trap for TCP connections (**tty**)
SNMP version	If no **version** keyword is present, the default is Version 1.
SNMPv3 authentication	If no keyword is entered, the default is the **noauth** (noAuthNoPriv) security level.
SNMP notification type	If no type is specified, all notifications are sent.

SNMP Configuration Guidelines

An SNMP *group* is a table that maps SNMP users to SNMP views. An SNMP *user* is a member of an SNMP group. An SNMP *host* is the recipient of an SNMP trap operation. An SNMP *engine ID* is a name for the local or remote SNMP engine.

When configuring SNMP, follow these guidelines:

- When configuring an SNMP group, do not specify a notify view. The **snmp-server host** global configuration command autogenerates a notify view for the user and then adds it to the group associated with that user. Modifying the group's notify view affects all users associated with that group. For information about when you should configure notify views, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*.
- To configure a remote user, specify the IP address or port number for the remote SNMP agent of the device where the user resides.
- Before you configure remote users for a particular agent, configure the SNMP engine ID, using the **snmp-server engineID** global configuration with the **remote** option. The remote agent's SNMP engine ID and user password are used to compute the authentication and privacy digests. If you do not configure the remote engine ID first, the configuration command fails.
- When configuring SNMP informs, you need to configure the SNMP engine ID for the remote agent in the SNMP database before you can send proxy requests or informs to it.
- If a local user is not associated with a remote host, the switch does not send informs for the **auth** (authNoPriv) and the **priv** (authPriv) authentication levels.
- Changing the value of the SNMP engine ID has important side effects. A user's password (entered on the command line) is converted to an MD5 or SHA security digest based on the password and the local engine ID. The command-line password is then destroyed, as required by RFC 2274. Because of this deletion, if the value of the engine ID changes, the security digests of SNMPv3 users become invalid, and you need to reconfigure SNMP users by using the **snmp-server user** *username* global configuration command. Similar restrictions require the reconfiguration of community strings when the engine ID changes.

Disabling the SNMP Agent

To disable the SNMP agent, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# no snmp-server`	Disables the SNMP agent operation.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show running-config`	Verifies your entries.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

The **no snmp-server** global configuration command disables all running versions (Version 1, Version 2C, and Version 3) on the device. No specific IOS command exists to enable SNMP. The first **snmp-server** global configuration command that you enter enables all versions of SNMP.

Configuring Community Strings

You use the SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to permit access to the agent on the switch. Optionally, you can specify one or more of these characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent
- A MIB view, which defines the subset of all MIB objects accessible to the given community
- Read and write or read-only permission for the MIB objects accessible to the community

To configure a community string on the switch, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# [no] snmp-server community string [view view-name] [ro | rw] [access-list-number]`	Configures the community string. • For *string*, specify a string that acts like a password and permits access to the SNMP protocol. You can configure one or more community strings up to 117 characters. • (Optional) For **view**, specify the view record accessible to the community. • (Optional) Specify either read-only (**ro**) if you want authorized management stations to retrieve MIB objects, or specify read-write (**rw**) if you want authorized management stations to retrieve and modify MIB objects. By default, the community string permits read-only access to all objects. • (Optional) For *access-list-number*, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999. To remove a specific community string, use the **no snmp-server community** *string* global configuration command.
Step 3	`Switch(config)# access-list access-list-number {deny | permit} source [source-wildcard]`	(Optional) If you specified an IP standard access list number in Step 2, then create the list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. • For *source*, enter the IP address of the SNMP managers that are permitted to use the community string to gain access to the agent. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.
Step 4	`Switch(config)# end`	Return to privileged EXEC mode.
Step 5	`Switch# show running-config`	Verifies your entries.
Step 6	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Note To disable access for an SNMP community, set the community string for that community to the null string (do not enter a value for the community string).

Note The **snmp-server enable informs** command is not supported. To enable the sending of SNMP inform notifications, use the **snmp-server enable traps** command combined with the **snmp-server host** *host-addr* **informs** command.

This example shows how to assign the string *comaccess* to SNMP, to allow read-only access, and to specify that IP access list 4 can use the community string to gain access to the switch SNMP agent:

```
Switch(config)# snmp-server community comaccess ro 4
```

Configuring SNMP Groups and Users

You can specify an identification name (engine ID) for the local or remote SNMP server engine on the switch. You can configure an SNMP server group that maps SNMP users to SNMP views, and you can add new users to the SNMP group.

To configure SNMP on the switch, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# snmp-server engineID {local engineid-string | remote ip-address [udp-port port-number] engineid-string}`	Configures a name for either the local or remote copy of SNMP. • The *engineid-string* is a 24-character ID string with the name of the copy of SNMP. You need not specify the entire 24-character engine ID if it has trailing zeros. Specify only the portion of the engine ID up to the point where only zeros remain in the value. For example, to configure an engine ID of 123400000000000000000000, you can enter this: **snmp-server engineID local 1234** • If you select **remote**, specify the *ip-address* of the device that contains the remote copy of SNMP and the optional UDP port on the remote device. The default is 162.

	Command	Purpose
Step 3	`Switch(config)#` **`snmp-server group`** *`groupname`* {**`v1`** / **`v2c`** | **`v3`** [**`auth`** | **`noauth`** | **`priv`**]} [**`read`** *`readview`*] [**`write`** *`writeview`*] [**`notify`** *`notifyview`*] [**`access`** *`access-list`*]	Configures a new SNMP group on the remote device. • For *groupname,* specify the name of the group. • Specify a security model: – **v1** is the least secure of the possible security models. – **v2c** is the second least secure model. It allows transmission of informs and integers twice the normal width. – **v3,** the most secure, requires you to select an authentication level: **auth**—Enables the Message Digest 5 (MD5) and the Secure Hash Algorithm (SHA) packet authentication. **noauth**—The noAuthNoPriv security level. This is the default if no keyword is specified. **priv**—Enables Data Encryption Standard (DES) packet encryption (also called *privacy*). **Note** The **priv** keyword is available only when the crypto software image is installed. • (Optional) Enter **read** *readview* with a string (not to exceed 64 characters) that is the name of the view in which you can only view the contents of the agent. • (Optional) Enter **write** *writeview* with a string (not to exceed 64 characters) that is the name of the view in which you enter data and configure the contents of the agent. • (Optional) Enter **notify** *notifyview* with a string (not to exceed 64 characters) that is the name of the view in which you specify a notify, inform, or trap. • (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.

	Command	Purpose
Step 4	`Switch(config)# snmp-server user username groupname [remote host [udp-port port]] {v1 / v2c | v3 [auth {md5 | sha} auth-password]} [encrypted] [access access-list]`	Configures a new user to an SNMP group. • The *username* is the name of the user on the host that connects to the agent. • The *groupname* is the name of the group to which the user is associated. • (Optional) Enter **remote** to specify a remote SNMP entity to which the user belongs and the hostname or IP address of that entity with the optional UDP port number. The default is 162. • Enter the SNMP version number (**v1**,or **v2c**, or **v3**). If you enter **v3**, you have these additional options: – **auth** is an authentication level setting session, which can be either the HMAC-MD5-96 or the HMAC-SHA-96 authentication level, and requires a password string (not to exceed 64 characters). – **encrypted** specifies that the password appears in encrypted format. • (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.
Step 5	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 6	`Switch# show running-config`	Verifies your entries.
Step 7	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring SNMP Notifications

A trap manager is a management station that receives and processes traps. Traps are system alerts that the switch generates when certain events occur. By default, no trap manager is defined, and no traps are sent. Switches running IOS Cisco IOS Release 12.2(31)SG can have an unlimited number of trap managers.

Note Many commands use the word *traps* in the command syntax. Unless there is an option in the command to select either traps or informs, the keyword *traps* refers to either traps, informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

Table 51-3 describes the supported switch traps (notification types). You can enable any or all of these traps and configure a trap manager to receive them.

Table 51-3 Switch Notification Types

Notification Type Keyword	Description
bgp	Generates BGP state change traps. **Note** This option is only available when the enhanced multilayer image is installed.
bridge	Generates STP bridge MIB traps.
config	Generates a trap for SNMP configuration changes.
config-copy	Generates a trap for SNMP copy configuration changes.
cpu	Allows cpu-related traps.
eigrp	Enable BGP traps. **Note** This option is only available when the enhanced multilayer image is installed.
entity	Generates a trap for SNMP entity changes.
envmon	Generates environmental monitor traps. You can enable any or all of these environmental traps: fan, shutdown, supply, temperature.
flash	Generates SNMP FLASH notifications.
fru-ctrl	Enable SNMP entity FRU control traps.
hsrp	Generates a trap for Hot Standby Router Protocol (HSRP) changes.
ipmulticast	Generates a trap for IP multicast routing changes.
isis	Enable IS-IS traps. **Note** This option is only available when the enhanced multilayer image is installed.
mac-notification	Generates a trap for MAC address notifications.
msdp	Generates a trap for Multicast Source Discovery Protocol (MSDP) changes. **Note** This option is only available when the enhanced multilayer image is installed.
ospf	Generates a trap for Open Shortest Path First (OSPF) changes. You can enable any or all of these traps: Cisco specific, errors, link-state advertisement, rate limit, retransmit, and state changes. **Note** This option is only available when the enhanced multilayer image is installed.
pim	Generates a trap for Protocol-Independent Multicast (PIM) changes. You can enable any or all of these traps: invalid PIM messages, neighbor changes, and rendezvous point (RP)-mapping changes.
port-security	Generates SNMP port security traps. You can also set a maximum trap rate per second. The range is from 0 to 1000; the default is 0, which means that there is no rate limit.
rf	Enable all SNMP traps defined in Cisco-RF-MIB.
snmp	Generates a trap for SNMP-type notifications for authentication, cold start, warm start, link up or link down.

Table 51-3 Switch Notification Types (continued)

Notification Type Keyword	Description
storm-control	Generates a trap for SNMP storm-control. You can also set a maximum trap rate per second. The range is from 0 to 1000; the default is 0 (no limit is imposed; a trap is sent at every occurrence).
stpx	Generates SNMP STP Extended MIB traps.
syslog	Generates SNMP syslog traps.
tty	Generates a trap for TCP connections. This trap is enabled by default.
vlan-membership	Generates a trap for SNMP VLAN membership changes.
vlancreate	Generates SNMP VLAN created traps.
vlandelete	Generates SNMP VLAN deleted traps.
vtp	Generates a trap for VLAN Trunking Protocol (VTP) changes.

You can use the **snmp-server host** global configuration command to a specific host to receive the notification types listed in Table 51-3.

To configure the switch to send traps or informs to a host, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	`Switch(config)# snmp-server engineID remote ip-address engineid-string`	Specifies the engine ID for the remote host.
Step 3	`Switch(config)# snmp-server user username groupname remote host [udp-port port] {v1 / v2c | v3 [auth {md5 | sha} auth-password]} [encrypted] [access access-list]`	Configures an SNMP user to be associated with the remote host created in Step 2. **Note** You cannot configure a remote user for an address without first configuring the engine ID for the remote host. If you try to configure the user before configuring the remote engine ID, you receive an error message, and the command is not executed.

	Command	Purpose
Step 4	`Switch(config)# snmp-server host host-addr [traps | informs] [version {1 / 2c | 3 [auth | noauth | priv]}] community-string [udp-port port] [notification-type]`	Specifies the recipient of an SNMP trap operation. • For *host-addr,* specify the name or Internet address of the host (the targeted recipient). • (Optional) Enter **traps** (the default) to send SNMP traps to the host. • (Optional) Enter **informs** to send SNMP informs to the host. • (Optional) Specify the SNMP **version** (**1**, **2c**, or **3**). SNMPv1 does not support informs. • (Optional) For Version 3, select authentication level **auth, noauth**, or **priv.** **Note** The **priv** keyword is available only when the crypto software image is installed. • For *community-string*, enter the password-like community string sent with the notification operation. • (Optional) For **udp-port** *port*, enter the remote device UDP port. • (Optional) For *notification-type*, use the keywords listed in Table 51-3 on page 51-12. If no type is specified, all notifications are sent.
Step 5	`Switch(config)# snmp-server enable traps notification-types`	Enables the switch to send traps or informs and specify the type of notifications to be sent. For a list of notification types, see Table 51-3 on page 51-12, or enter this: **snmp-server enable traps ?** To enable multiple types of traps, you must enter a separate **snmp-server enable traps** command for each trap type.
Step 6	`Switch(config)# snmp-server trap-source interface-id`	(Optional) Specifies the source interface, which provides the IP address for the trap message. This command also sets the source IP address for informs.
Step 7	`Switch(config)# snmp-server queue-length length`	(Optional) Establishes the message queue length for each trap host. The range is 1 to 1000; the default is 10.
Step 8	`Switch(config)# snmp-server trap-timeout seconds`	(Optional) Defines how often to resend trap messages. The range is 1 to 1000; the default is 30 seconds.
Step 9	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 10	`Switch# show running-config`	Verifies your entries.
Step 11	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

The **snmp-server host** command specifies which hosts receive the notifications. The **snmp-server enable trap** command globally enables the mechanism for the specified notification (for traps and informs). To enable a host to receive an inform, you must configure an **snmp-server host informs** command for the host and globally enable informs by using the **snmp-server enable traps** command.

To remove the specified host from receiving traps, use the **no snmp-server host** *host* global configuration command. The **no snmp-server host** command with no keywords disables traps, but not informs, to the host. To disable informs, use the **no snmp-server host informs** global configuration command. To disable a specific trap type, use the **no snmp-server enable traps** *notification-types* global configuration command.

Setting the Agent Contact and Location Information

To set the system contact and location of the SNMP agent so that these descriptions can be accessed through the configuration file, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **snmp-server contact** *text*	Sets the system contact string. For example: **snmp-server contact Dial System Operator at beeper 21555.**
Step 3	Switch(config)# **snmp-server location** *text*	Sets the system location string. For example: **snmp-server location Building 3/Room 222**
Step 4	Switch(config)# **end**	Returns to privileged EXEC mode.
Step 5	Switch# **show running-config**	Verifies your entries.
Step 6	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Limiting TFTP Servers Used Through SNMP

To limit the TFTP servers used for saving and loading configuration files through SNMP to the servers specified in an access list, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **snmp-server tftp-server-list** *access-list-number*	Limits TFTP servers used for configuration file copies through SNMP to the servers in the access list. For *access-list-number*, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.
Step 3	Switch(config)# **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. • For *source*, enter the IP address of the TFTP servers that can access the switch. • (Optional) For *source-wildcard*, enter the wildcard bits, in dotted decimal notation, to be applied to the source. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.

	Command	Purpose
Step 4	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# show running-config`	Verifies your entries.
Step 6	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

SNMP Examples

This example shows how to enable all versions of SNMP. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public*. This configuration does not cause the switch to send any traps.

```
Switch(config)# snmp-server community public
```

This example shows how to permit any SNMP manager to access all objects with read-only permission using the community string *public*. The switch also sends VTP traps to the hosts 192.180.1.111 and 192.180.1.33 using SNMPv1 and to the host 192.180.1.27 using SNMPv2C. The community string *public* is sent with the traps.

```
Switch(config)# snmp-server community public
Switch(config)# snmp-server enable traps vtp
Switch(config)# snmp-server host 192.180.1.27 version 2c public
Switch(config)# snmp-server host 192.180.1.111 version 1 public
Switch(config)# snmp-server host 192.180.1.33 public
```

This example shows how to allow read-only access for all objects to members of access list 4 that use the *comaccess* community string. No other SNMP managers have access to any objects. SNMP Authentication Failure traps are sent by SNMPv2C to the host *cisco.com* using the community string *public*.

```
Switch(config)# snmp-server community comaccess ro 4
Switch(config)# snmp-server enable traps snmp authentication
Switch(config)# snmp-server host cisco.com version 2c public
```

This example shows how to send Entity MIB traps to the host *cisco.com*. The community string is restricted. The first line enables the switch to send Entity MIB traps in addition to any traps previously enabled. The second line specifies the destination of these traps and overwrites any previous **snmp-server host** commands for the host *cisco.com*.

```
Switch(config)# snmp-server enable traps entity
Switch(config)# snmp-server host cisco.com restricted entity
```

This example shows how to enable the switch to send all traps to the host *myhost.cisco.com* using the community string *public*:

```
Switch(config)# snmp-server enable traps
Switch(config)# snmp-server host myhost.cisco.com public
```

This example shows how to associate a user with a remote host and to send **auth** (authNoPriv) authentication-level informs when the user enters global configuration mode:

```
Switch(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b
Switch(config)# snmp-server group authgroup v3 auth
Switch(config)# snmp-server user authuser authgroup remote 192.180.1.27 v3 auth md5
mypassword
Switch(config)# snmp-server user authuser authgroup v3 auth md5 mypassword
Switch(config)# snmp-server host 192.180.1.27 informs version 3 auth authuser config
Switch(config)# snmp-server enable traps
```

```
Switch(config)# snmp-server inform retries 0
```

Displaying SNMP Status

To display SNMP input and output statistics, including the number of illegal community string entries, errors, and requested variables, use the **show snmp** privileged EXEC command.You can also use the other privileged EXEC commands in Table 51-4 to display SNMP information. For information about the fields in the output displays, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.4.*

Table 51-4 Commands for Displaying SNMP Information

Feature	Default Setting
show snmp	Displays SNMP statistics.
show snmp engineID	Displays information on the local SNMP engine and all remote engines that have been configured on the device.
show snmp group	Displays information on each SNMP group on the network.
show snmp pending	Displays information on pending SNMP requests.
show snmp sessions	Displays information on the current SNMP sessions.
show snmp user	Displays information on each SNMP user name in the SNMP users table.

Note The **snmp-server enable informs** command is not supported. To enable the sending of SNMP inform notifications, use the **snmp-server enable traps** command combined with the **snmp-server host** *host-addr* **informs** command.